可逆背包

问题简介

考虑这样一个问题：有 n 个物品，体积分别是 $w_1,w_2,\cdots,w_n$，背包容量为 m，共有 n 次询问，第 i 次询问要求不选物品 i 时共有多少选法装满背包。

思路分析

如果对于每个询问依次求 01 背包问题，单次询问需要 $O(nm)$ 的时间，总复杂度 $O(n^2m)$。而退背包能让我们不需要对于每个询问都进行完整的 DP，而是用 $O(m)$ 时间去掉选择了物品 i 的方案数，那么整个过程需要一次 $O(nm)$ 的DP和 n 次 $O(m)$ 的退背包/加背包过程，整体复杂度就降到了 $O(nm)$。

算法步骤

1. 执行一次 01 背包算法过程

```
for(int j=m;j>=w[i];--j)
    f[j]+=f[j-w[i]];
```

2. 对每个物品，将其看作是最后一个加入背包的物品，然后减掉它的方案数

```
memcpy(g,f,sizeof f);
for(int j=w[i];j<=m;++j)
    g[j]-=g[j-w[i]];
```

例题

例 1

洛谷 P4141 消失之物

题意

有 n 个物品，体积分别是 $w_1,w_2,\cdots,w_n$。现第 i 个物品丢失，用剩下的 $n-1$ 个物品装满容积为 x 的背包，有几种方法。把答案记为 $cnt(i,x)$，要得到所有 $i\in[1,n],x\in[1,m]$ 的 $cnt(i,x)$ 表格。输出 $n\times m$ 的矩阵，表示 $cnt(i,x)$ 的末位数字。

$1\le n,m\le 2000$。

题解

令 $f[j][0]$ 表示不算消失的物品组成容积为 j 的方案数，即 01 背包问题，令 $f[j][1]$ 表示删掉某个物品后组成容积为 j 的方案数。初始化是 $f[0][0]=f[0][1]=1$，即容积为 0 时方案数为 1

```
#include<bits/stdc++.h>
using namespace std;
const int N=2e3+5;
int f[N][2],w[N];

int main(){
    int n,m;
    scanf("%d %d",&n,&m);
    for(int i=1;i<=n;i++) scanf("%d",&w[i]);
    f[0][0]=f[0][1]=1;
    for(int i=1;i<=n;i++){
        for(int j=m;j>=w[i];j--){
            f[j][0]=(f[j][0]+f[j-w[i]][0])%10;
        }
    }

    for(int i=1;i<=n;i++){
        for(int j=1;j<=m;j++){
            if(j-w[i]>=0) f[j][1]=(f[j][0]-f[j-w[i]][1]+10)%10;
            else f[j][1]=f[j][0]%10;
            printf("%d",f[j][1]);
        }
        puts("");
    }
    return 0;
}
```

例 2

CF 1111 D. Destroy the Colony

题意

给定一个长度为偶数的由大小写字母组成的字符串 s，一个“好”字符串可通过重排字母顺序使得所有相同字母的位置都在 s 的同一半（$[1,n/2] or [n/2+1,n]$）来生成，q 个询问，给定一对 (i,j)，要求字母 $s[i],s[j]$ 也在 s 的同一半的“好”字符串数量。

题解

注意到询问本质上最多只有 52^2 种可能，则考虑先把所有询问算出来，最后 $O(1)$ 回答询问。

1. 令 k 为字母种数，统计各字母的出现次数 $c_1,c_2,\cdots,c_k$□若有 $i_1,i_2,\cdots,i_p$ 使得 $c_{i_1}+c_{i_2}+\cdots+c_{i_p}=\dfrac{n}{2}$□则把它们放到第一组，剩下的放到第二组，第一组的排列数为 $\dfrac{(\dfrac{n}{2})!}{c_{i_1}!c_{i_2}!\cdots c_{i_p}!}$□第二组排列数为 $\dfrac{(\dfrac{n}{2})!}{c_{j_1}!c_{j_2}!\cdots c_{j_s}!}$□总方法数为 $W=\dfrac{(\dfrac{n}{2})!(\dfrac{n}{2})!}{c_1!c_2!\cdots c_k!}$□
2. 注意到 c_i 求和为 $\dfrac{n}{2}$ 的组合数可用 01 背包来求，在没有 (i,j) 限制下的“好”字符串数量为 $W*dp[n/2]*2$□
3. 现考虑 (i,j) 的限制，即字母 $s[i]$ 和 $s[j]$ 需在同一组中，等价于去掉字母 $s[i]$ 和 $s[j]$ 的 01 背包问题，即作了一次 01 背包问题后，再枚举 i,j 做退背包过程，将去掉 (i,j) 的 DP 值记作 $ans[i][j]$□
4. 最后对每个询问直接 $O(1)$ 输出答案即可。